(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 126 685 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
22.08.2001 Bulletin 2001/34

(51) Int Cl.[7]: **H04M 3/42**, H04Q 3/00

(21) Application number: **01300918.8**

(22) Date of filing: **01.02.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **09.02.2000 US 501134**

(71) Applicant: **Siemens Information and Communication Networks Inc.**
Boca Raton, FL 33487 (US)

(72) Inventor: **Hettish, Mark Bernard**
Cary, NC 27511 (US)

(74) Representative: **Wilding, Frances Ward et al**
Haseltine Lake & Co
Imperial House
15-19 Kingsway
London WC2B 6UD (GB)

(54) System and method for reporting the addition and deletion of TAPI line and phone devices in absence of such notification from a PBX

(57) A TAPI service provider may request configuration information from hardware such as a PBX. Configuration information received from the hardware is stored in a list (C2) and compared to the existing configuration list (C1). Two new lists are then generated: an added list (C4) and a deleted list (C3). All devices in the deleted list are deleted from the internal database and deleted messages are sent to the TAPI service provider. All new devices are temporarily added to the internal database and addition messages are sent to the TAPI service provider. The new device configuration list is then saved, and the added devices are marked as permanent.

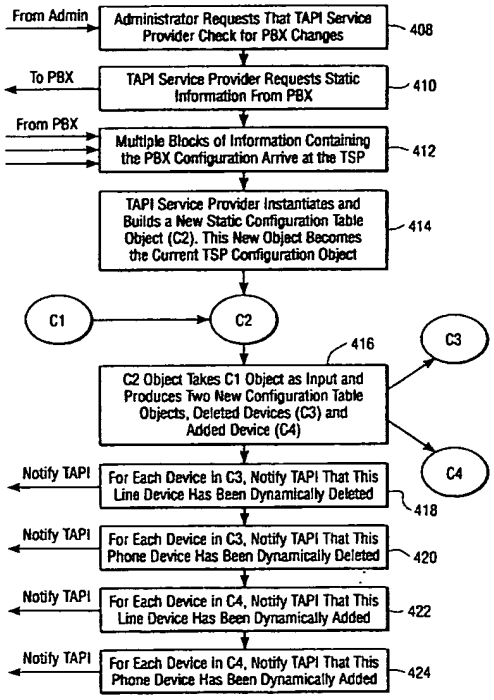

FIG. 4B

EP 1 126 685 A2

Description

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the official file or records available to the public, but otherwise reserves all copyright rights whatsoever.

[0002] The present invention relates to communications systems and, in particular, to a communication system employing a private branch exchange (PBX) and a TAPI interface.

[0003] The Telephony Application Programming Interface (TAPI) is a high level programming interface for Windows™ which supports many types of telephony applications associated with conventional analog public telephone lines, PBX phone lines, ISDN phone lines, and the like. Thus, TAPI allows a communication application to support numerous telephony operations through a variety of mediums by making a function call to TAPI which will drive the hardware (fax/modem card, DSP card, network switch, and the like) coupled thereto.

[0004] The TAPI architecture 100 is illustrated in FIG. 1. As shown, the TAPI architecture 100 includes a TAPI implementation 104 interfaced to telephony'application programs 102. TAPI 104 provides a connection to a TAPI service provider, such as a TAPI server 106, which then interfaces to hardware such as voices cards 108a, H.323 interfaces 108b, or PBX'S 108c.

[0005] The TAPI specification requires that device configuration information be available at the startup of the TAPI service provider. If unsolicited addition and deletion events from a telephony device hardware, such as a PBX, are not supported (i.e., the hardware does not automatically inform the TAPI service provider of updates), the TAPI service provider can only do configuration change updates at start up. The TAPI service provider does so by requesting the static device configuration from the PBX, building the initial internal database and reporting all the devices to the TAPI. If changes are made on the PBX, the telephony server system administrator would be required to shut down the TAPI service provider and restart it again. This has the disadvantage of disrupting service to all users of the TAPI service provider. Moreover, in such a system, there is no way to shut down a TAPI service provider if it is in use. It will be shut down only if every TAPI application using it shuts down. Since several thousand clients can be supported, this can provide severe disadvantages in administration.

[0006] These and other drawbacks in the prior art may be overcome in large part or at least mitigated by a system and method for client configuration in a TAPI environment according to embodiments of the present invention.

[0007] The invention is defined in the independent claims, to which reference should now be made. Further advantageous features are detailed in the dependent claims.

[0008] Briefly, embodiments of the present invention provide a method whereby a TAPI service provider may request configuration information from hardware such as a PBX. Configuration information received from the hardware may be stored in a list and compared to the existing configuration list. Two new lists are then generated: an added list and a deleted list. All devices in the deleted list are deleted from the internal database and deleted messages are sent to the TAPI service provider. All new devices are temporarily added to the internal database and addition messages are sent to the TAPI service provider. The new device configuration list is then saved, and the added devices are marked as permanent.

[0009] A better understanding of the invention is obtained when the following detailed description of embodiments thereof is considered in conjunction with the following drawings in which:

FIG. 1 is a diagram representative of the TAPI architecture;

FIG. 2 is a diagram illustrating a computer system employing a TAPI system according to an implementation of the present invention;

FIG. 3 is a block diagram of the computer system of FIG. 2 according to an implementation of the present invention; and

FIG. 4A and FIG. 4B are flowcharts illustrating operation of an implementation of the invention.

[0010] FIGS. 2-4 illustrate an improved system and method for a TAPI service provider to request and update configuration information from hardware such as a PBX. Configuration information received from the hardware is stored in a list and compared to the existing configuration list. Two new lists are then generated: an added list and a deleted list. All devices in the deleted list are deleted from the internal database and deleted messages are sent to the TAPI service provider. All new devices are temporarily added to the internal database and addition messages are sent to the TAPI service provider. The new device configuration list is then saved, and the added devices are marked as permanent.

[0011] An exemplary TAPI client 202 is shown in FIG. 2. The TAPI client 202 may be embodied as a personal computer, including a system unit 11, a keyboard 12, a mouse 13, and a display 140. Also shown are one or more speakers 150a, 150b, and a microphone 1601. The screen 160 of the display device 14 is used to present a graphical user interface (GUI) and particularly, a TAPI client window 3008. The graphical user interface supported by the operating system allows the user to employ a point and click method of input, i.e., by moving the mouse pointer or cursor (not shown) to an icon representing a data object at a particular location on the screen 160 and pressing one or more of the mouse but-

tons to perform a user command or selection. The GUI may be any of the Windows GUIs available from Microsoft Corporation or the Macintosh OS, available from Apple Computer.

[0012] FIG. 3 shows a block diagram of the components of the personal computer shown in FIG. 2. The system unit 11 includes a system bus or a plurality of system buses 21 to which various components are coupled and by which communication between the various components is accomplished. The microprocessor 22 is coupled to the system bus 21 and is supported by the read only memory (ROM) 23 and the random access memory (RAM) 24 also connected to the system bus 21. The microprocessor 22 may be embodied as any of a variety of microprocessors, including Intel x86, Pentium or Pentium II or compatible processors.

[0013] The ROM 23 contains among other code the basic input output system (BIOS) which controls basic hardware operations such as the interaction of the disk drives and the keyboard. The RAM 24 is the main memory into which the operating system and applications programs are loaded. The memory management chip 25 is connected to the system bus 21 and controls direct memory access operations including passing data between the RAM 24 and hard disk drive 26 and floppy disk drive 27. A CD ROM drive (or DVD or other optical drive) 32 may also be coupled to the system bus 21 and is used to store a large amount of data, such as a multimedia program or a large database.

[0014] Also connected to the system bus 21 are various I/O controllers: The keyboard controller 28, the mouse controller 29, the video controller 30, and the audio controller 31. The keyboard controller 28 provides the hardware interface for the keyboard; the mouse controller 29 provides the hardware interface for the mouse 13; the video controller 30 is the hardware interface for the video display 14; and the audio controller 31 is the hardware interface for the speakers 15 and microphone 16. The speakers 150a, b and the microphone 1601 allow for audio communication during telephony operation. In operation, keyboard strokes are detected by the keyboard controller 28 and corresponding signals are transmitted to the microprocessor 22; similarly, mouse movements and button clicks are detected by the mouse controller and provided to the microprocessor 22. Typically, the keyboard controller 28 and the mouse controller 29 assert interrupts at the microprocessor 22. In response, the microprocessor 22 executes a corresponding interrupt routine, as is known. Additionally, an interrupt controller (not shown) may be provided to arbitrate among interrupt requests. An I/O controller or network interface 40 enables communication over a network 46, such as a packet network.

[0015] One embodiment of the present invention is as a set of instructions in a code module resident in the RAM 24. Until required by the computer system, the set of instructions may be stored in another computer memory, such as the hard disk 26, on an optical disk for use in the CD ROM drive 32, or a floppy disk for use in the floppy disk drive 27.

[0016] As shown in the figure, the operating system 50, the TAPI application 52, the TAPI service provider 53, and one or more configuration table objects 56 are resident in the RAM 24. As is known, the operating system 50 functions to generate a graphical user interface on the display 14. The TAPI application program 52 performs TAPI functionality, including generation of a TAPI client window 3008 (FIG. 2) in the GUI. The TAPI service provider 53 implements an interface to the hardware, as will be described in greater detail below.

[0017] Turning now to FIG. 4A and 4B, a flowchart illustrating operation of an implementation of the invention is shown. Upon startup, in a step 402, the TAPI service provider 53 requests static configuration information from the PBX (not shown) or other telephony hardware. Such static configuration information can be static line or static phone device information. In the case of a line device, the configuration information includes the device ID and the number of addresses on the line. In the case of a phone device, the information includes the device ID. Turning back to FIG. 4A, in a step 404, the PBX returns multiple blocks of information of PBX configuration information to the TAPI service provider 53. Next, in a step 406, the TAPI service provider 53 instantiates and builds a static configuration table object (C1).

[0018] In normal operation, devices are added or deleted to the PBX without notifications being sent to the TAPI service provider 53. Turning now to FIG. 4B, in a step 408, the administrator may request the TAPI service provider 53 to check for PBX configuration changes. In a step 410, the TAPI service provider 53 requests static information from the PBX. In a step 412, multiple blocks of information containing the PBX configuration data arrive at the TAPI service provider 53. In a step 414, the TAPI service provider 53 instantiates and builds a new static configuration table object (C2). This new object becomes the current TAPI configuration object. In a step 416, the TAPI service provider 53 takes the C2 object as an input and produces two new configuration table objects: deleted objects (C3) and added devices (C4).

[0019] In a step 418, for each device in C3, the TAPI application 52 is notified that the corresponding line device has been deleted. In a step 420, for each device in C3, the TAPI application 52 is notified that the corresponding phone device has been deleted. In a step 422, for each device in C4, the TAPI application 52 is notified that the corresponding line device has been added. Finally, in a step 424, for each device in C4, the TAPI application 52 is notified that the corresponding phone device has been added.

[0020] It is noted that the invention is not limited to the specific form set forth herein, but includes such alternatives, modifications and equivalents as can reasonably be included within the scope of the appended claims.

Claims

1. A method for updating configuration information in a TAPI system, comprising the steps of:

   storing configuration information in a first configuration table object (C1) ;
   requesting update static configuration information from a telephony device;
   receiving said update static configuration information; and
   storing said static update configuration information for system use as a second configuration table object (C2).

2. A method in accordance with claim 1, further comprising generating a deleted devices configuration table object (C3) and/or an added devices configuration table object (C4).

3. A method in accordance with claim 2, further comprising notifying said TAPI system that each device in said deleted devices configuration table object (C3) has been dynamically deleted and/or that each device in said added devices configuration table object (C4) has been dynamically added.

4. A method in accordance with any of the preceding claims wherein the configuration information is updated while the TAPI system is in use.

5. A method in accordance with any of the preceding claims wherein a system administrator requests a TAPI service provider to check for configuration changes.

6. A method in accordance with any of the preceding claims wherein the first configuration table object data is from configuration at start-up or from an earlier update.

7. A TAPI system, comprising:

   means (22) for storing configuration information in a first configuration table object;
   means (22) for requesting update static configuration information from a telephony device;
   means (22) for receiving said update static configuration information; and
   means (22) for storing said static update configuration information for system use as a second configuration table object.

8. A system in accordance with claim 7, said storing means (22) further comprising means (22) for generating a deleted devices configuration table object and/or an added devices configuration table object.

9. A system in accordance with claim 8, further comprising means (22) for notifying said TAPI system that each device in said deleted devices configuration table object has been dynamically deleted and/or means (22) for notifying said TAPI system that each device in said added devices configuration table object has been dynamically added.

10. A system in accordance with any of the preceding claims, wherein a TAPI service provider requests and updates configuration information from hardware.

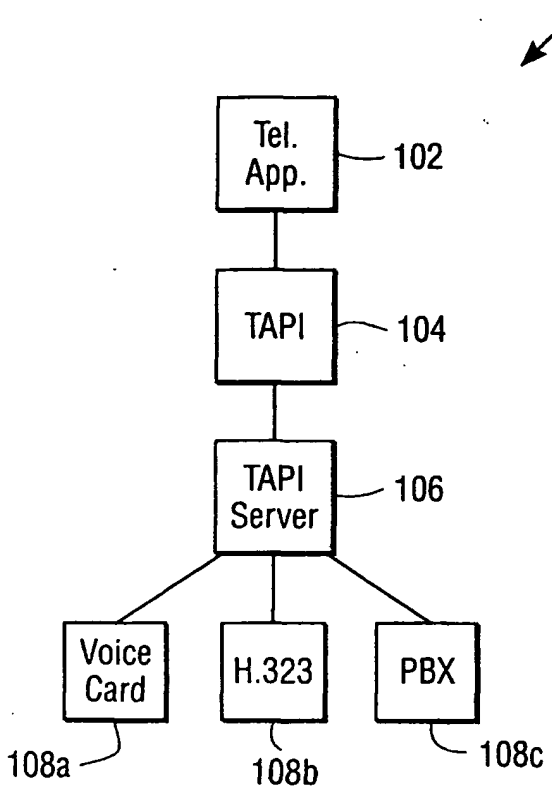
100
Tel. App.
102
TAPI
104
TAPI Server
106
Voice Card
H.323
PBX
108a
108b
108c

FIG. 1

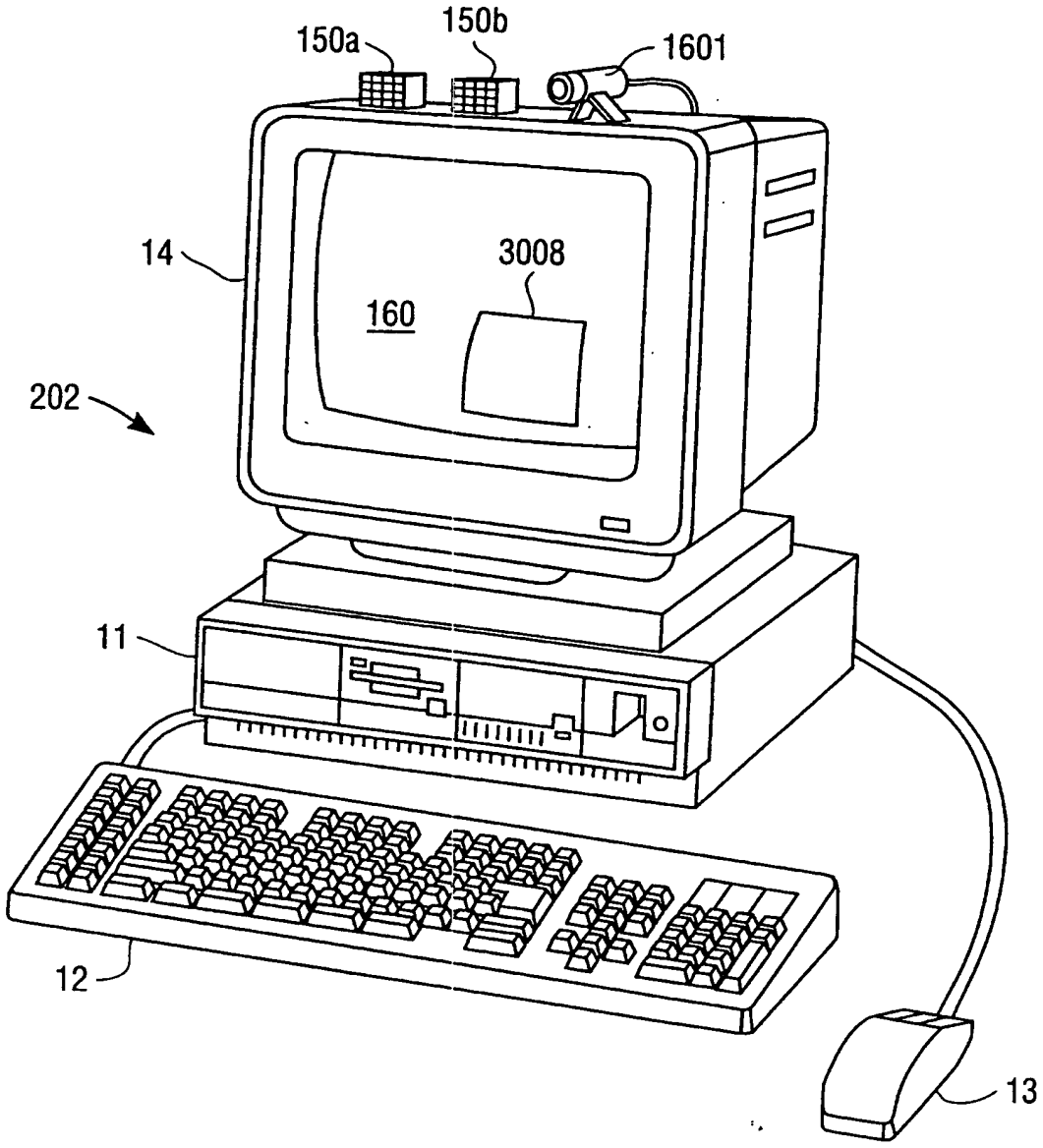

FIG. 2

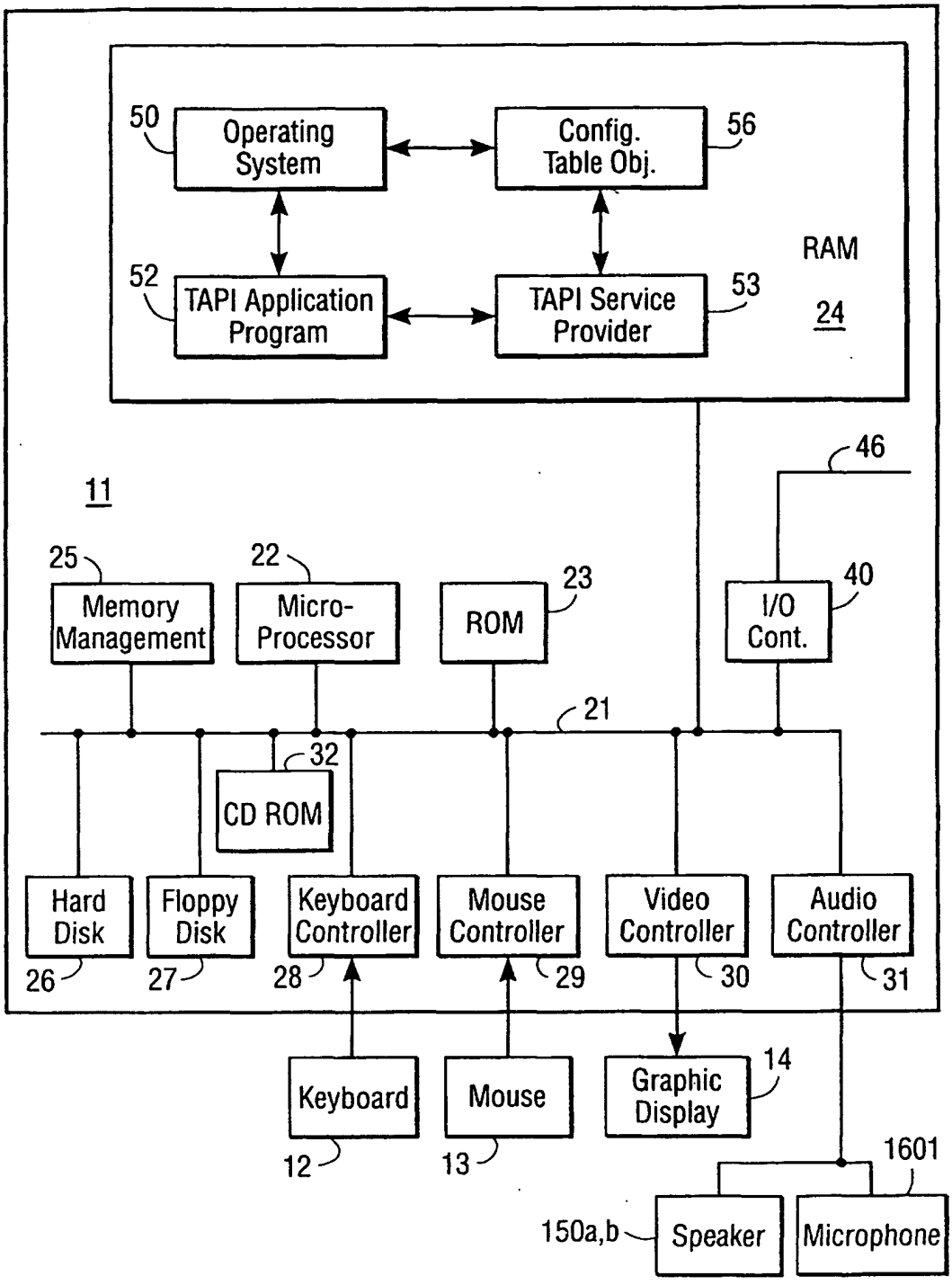
50
Operating System
56
Config. Table Obj.
RAM
24
52
TAPI Application Program
53
TAPI Service Provider
46
11
25
Memory Management
22
Micro-Processor
23
ROM
40
I/O Cont.
21
32
CD ROM
Hard Disk
Floppy Disk
Keyboard Controller
Mouse Controller
Video Controller
Audio Controller
26
27
28
29
30
31
Keyboard
Mouse
Graphic Display
14
12
13
1601
150a,b
Speaker
Microphone

FIG. 3

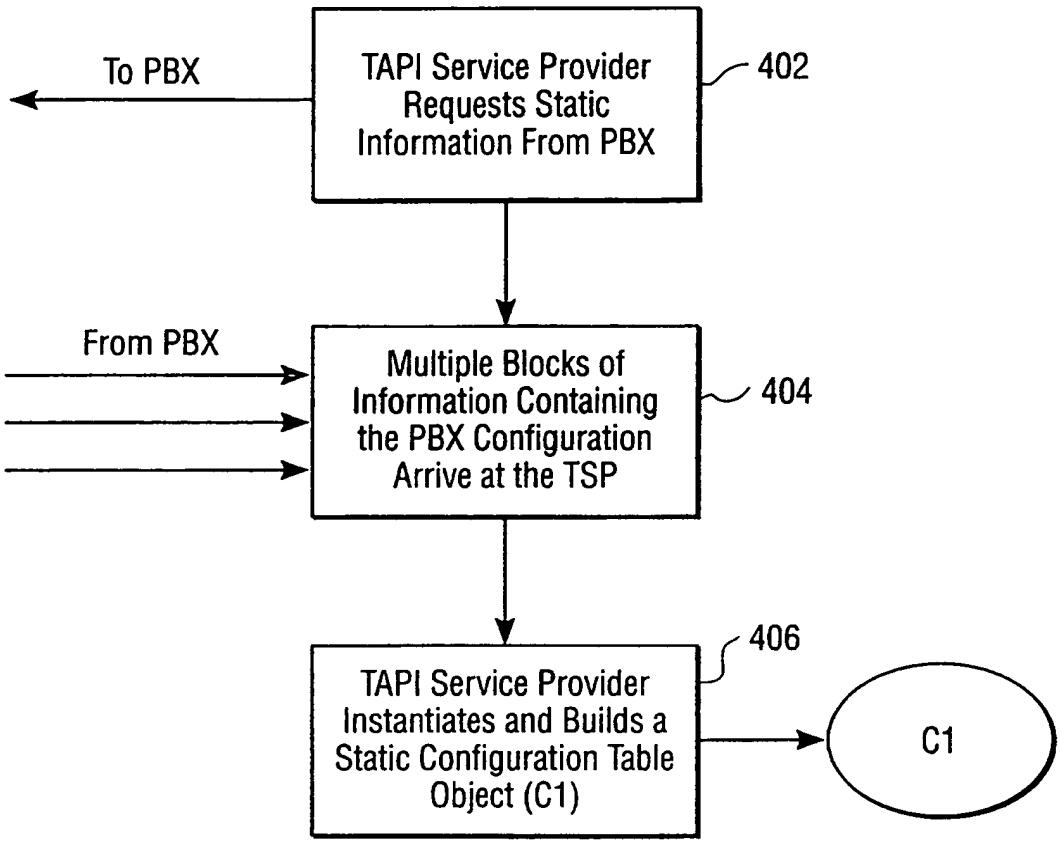
To PBX
TAPI Service Provider Requests Static Information From PBX
402
From PBX
Multiple Blocks of Information Containing the PBX Configuration Arrive at the TSP
404
406
TAPI Service Provider Instantiates and Builds a Static Configuration Table Object (C1)
C1

FIG. 4A

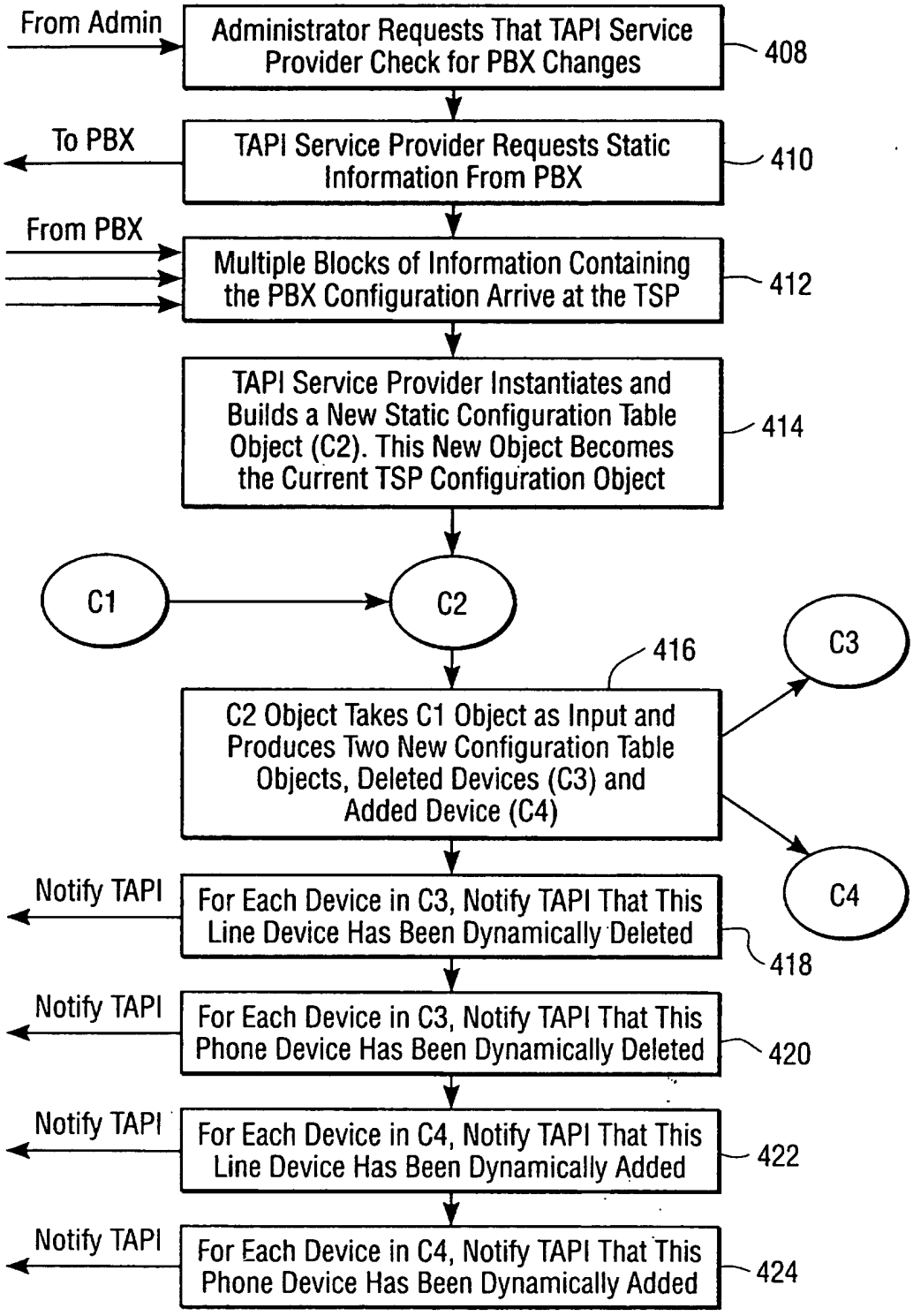
From Admin
Administrator Requests That TAPI Service Provider Check for PBX Changes
408
To PBX
TAPI Service Provider Requests Static Information From PBX
410
From PBX
Multiple Blocks of Information Containing the PBX Configuration Arrive at the TSP
412
TAPI Service Provider Instantiates and Builds a New Static Configuration Table Object (C2). This New Object Becomes the Current TSP Configuration Object
414
C1
C2
C3
416
C2 Object Takes C1 Object as Input and Produces Two New Configuration Table Objects, Deleted Devices (C3) and Added Device (C4)
C4
Notify TAPI
For Each Device in C3, Notify TAPI That This Line Device Has Been Dynamically Deleted
418
Notify TAPI
For Each Device in C3, Notify TAPI That This Phone Device Has Been Dynamically Deleted
420
Notify TAPI
For Each Device in C4, Notify TAPI That This Line Device Has Been Dynamically Added
422
Notify TAPI
For Each Device in C4, Notify TAPI That This Phone Device Has Been Dynamically Added
424

FIG. 4B